Iterative Approximate Byzantine Consensus under a *Generalized* Fault Model*

Lewis Tseng[1,3], and Nitin Vaidya[2,3]

[1] Department of Computer Science,
[2] Department of Electrical and Computer Engineering, and
[3] Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email: {ltseng3, nhv}@illinois.edu

Technical Report

May 21, 2012

Abstract

In this work, we consider a *generalized* fault model that can be used to represent a wide range of failure scenarios, including correlated failures and non-uniform node reliabilities. This fault model is general in the sense that fault models studied in prior related work, such as f-total and f-local models, are special cases of the generalized fault model. Under the generalized fault model, we explore iterative approximate Byzantine consensus (IABC) algorithms in arbitrary directed networks. We prove a necessary and sufficient condition for the existence of IABC algorithms. The use of the generalized fault model helps to gain a better understanding of IABC algorithms.

*This research is supported in part by National Science Foundation award CNS 1059540 and Army Research Office grant W-911-NF-0710287. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

1 Introduction

Dolev et al. [4] introduced the notion of *approximate Byzantine consensus* by relaxing the requirement of *exact* consensus [12]. The goal in approximate consensus is to allow the fault-free nodes to agree on values that are approximately equal to each other (and not necessarily exactly identical). In presence of Byzantine faults, while *exact* consensus is impossible in *asynchronous* systems [5], approximate consensus is achievable [4]. The notion of approximate consensus is of interest in *synchronous* systems as well, since approximate consensus can be achieved using distributed algorithms that do not require complete knowledge of the network topology [1]. The rest of the discussion in this paper assumes a *synchronous* systems.

The fault model assumed in much of the work on Byzantine consensus allows up to f Byzantine faulty nodes in the network. We will refer to this fault model as the "f-total" fault model [16, 10, 4, 12]. In prior work, other fault models have been explored as well. For instance, in the "f-local" fault model, up to f neighbors of *each* node in the network may be faulty [8, 2, 16], and in the f-fraction model [16], up to f fraction of the neighbors of each node may be faulty. In this paper, we consider a *generalized* fault model (to be described in the next section). The *generalized fault* model specifies a "fault domain", which is a collection of feasible fault sets (a similar fault model is recently presented in [9]). For example, in a system consisting of four nodes, namely, nodes $1, 2, 3$ and 4, the fault domain could be specified as $\mathcal{F} = \{\{1\}, \{2, 3, 4\}\}$. Thus, in this case, either node 1 may be faulty, or any subset of nodes in $\{2, 3, 4\}$ may be faulty. However, node 1 may not be faulty simultaneously with another node. The new fault model is general in the sense that the other fault models studied in the literature, such as f-total, f-local and f-fraction models, are special cases of the generalized fault model.

Analysis of consensus under the generalized fault model offers some new insights into how the choice of the fault model affects algorithm design. In particular, we consider "iterative" algorithms for achieving approximate Byzantine consensus in synchronous point-to-point networks that are modeled by arbitrary *directed* graphs. The *iterative approximate Byzantine consensus* (IABC) algorithms of interest have the following properties, which we will soon state more formally:

- *Initial state* of each node is equal to a real-valued *input* provided to that node.
- *Validity* condition: After each iteration of an IABC algorithm, the state of each fault-free node must remain in the *convex hull* of the states of the fault-free nodes at the end of the *previous* iteration.
- *Convergence* condition: For any $\epsilon > 0$, after a sufficiently large number of iterations, the states of the fault-free nodes are guaranteed to be within ϵ of each other.

This paper is a generalization of our recent work on IABC algorithms under the f-total fault model [14, 13]. The contributions of this paper are as follows:

- We identify a necessary condition on the communication graph for the existence of a correct IABC algorithm under the *generalized* fault model (Sections 3 and 4).
- We introduce a new IABC algorithm for the generalized fault model (Section 5) that uses only "local" information.
- A transition matrix representation of the new IABC algorithm is presented (Section 6). This representation is then used to prove the correctness of the proposed algorithm (Section 6.3).

Since the results here generalize our prior results [14, 13], naturally the proof techniques used here have some similarities to the prior work. The material in Section 6.3 bears the strongest similarity to our prior work. The rest of the paper, however, presents results that provide new intuition on the problem of approximate consensus. In particular, materials in Sections 4 and 5 shed light on how the fault model influences the design of IABC algorithms.

2 Models

Communication Model: The system is assumed to be *synchronous*. The communication network is modeled as a simple *directed* graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of n nodes, and $\mathcal{E}$ is the set of directed edges between the nodes in $\mathcal{V}$. We assume that $n \geq 2$, since the consensus problem for $n = 1$ is trivial. Node i can reliably transmit messages to node j if and only if the directed edge (i, j) is in $\mathcal{E}$. Each node can send messages to itself as well, however, for convenience, we exclude self-loops from set $\mathcal{E}$. That is, $(i, i) \notin \mathcal{E}$ for $i \in \mathcal{V}$. With a slight abuse of terminology, we will use the terms *edge* and *link* interchangeably in our presentation.

For each node i, let N_i^- be the set of nodes from which i has incoming edges. That is, $N_i^- = \{ j \mid (j, i) \in \mathcal{E} \}$. Similarly, define N_i^+ as the set of nodes to which node i has outgoing edges. That is, $N_i^+ = \{ j \mid (i, j) \in \mathcal{E} \}$. Nodes in N_i^- and N_i^+ are, respectively, said to be incoming and outgoing neighbors of node i. Since we exclude self-loops from $\mathcal{E}$, $i \notin N_i^-$ and $i \notin N_i^+$. However, we note again that each node can indeed send messages to itself.

Generalized Byzantine Failure Model: We consider the Byzantine failure model, with possible faulty nodes specified using a "fault domain" $\mathcal{F}$ (defined below). A faulty node may *misbehave* arbitrarily. Possible misbehavior includes transmitting incorrect and mismatching (or inconsistent) messages to different neighbors. The faulty nodes may collaborate with each other. Moreover, the faulty nodes are assumed to have a complete knowledge of the execution of the algorithm, including the states of all the nodes, the algorithm specification, and the network topology.

The generalized fault model is characterized using *fault domain* $\mathcal{F} \subseteq 2^{\mathcal{V}}$ as follows: Nodes in set F may fail during an execution of the algorithm only if there exists set $F^* \in \mathcal{F}$ such that $F \subseteq F^*$. Set F is then said to be a *feasible* fault set.

Definition 1 *Set $F \subseteq \mathcal{V}$ is said to be a feasible fault set, if there exists $F^* \in \mathcal{F}$ such that $F \subseteq F^*$.*

Thus, each set in $\mathcal{F}$ specifies nodes that may all potentially fail during a single execution of the algorithm (a similar fault model is also considered in [9]). This feature can be used to capture the notion of correlated failures. For example, consider a system consisting of four nodes, namely, nodes 1, 2, 3, and 4. Suppose that

$$\mathcal{F} = \{ \{1\}, \{2\}, \{3, 4\} \}$$

This definition of $\mathcal{F}$ implies that during an execution either (i) node 1 may fail, or (ii) node 2 may fail, or (iii) any subset of $\{3, 4\}$ may fail, and no other combination of nodes may fail (e.g., nodes 1 and 3 cannot both fail in a single execution). In this case, the reason that the set $\{3, 4\}$ is in the fault domain may be that the failures of nodes 3 and 4 are correlated.

The generalized fault model is also useful to capture variations in node reliability. For instance, in the above example, nodes 1 and 2 may be more reliable than nodes 3 and 4. Therefore, while

simultaneous failure of nodes 3 and 4 may occur, simultaneous failure of nodes 1 and 2 is less likely. Therefore, $\{1, 2\} \notin \mathcal{F}$.

Local knowledge of $\mathcal{F}$: To implement our IABC Algorithm presented in Section 5, it is sufficient for each node i to know $N_i^- \cap F$, for each feasible fault set F. In other words, each node only needs to know the set of its incoming neighbors that may fail simultaneously. Thus, the iterative algorithm can be implemented using only "local" information regarding $\mathcal{F}$.

3 Iterative Approximate Byzantine Consensus (IABC) Algorithms

In this section, we describe the structure of the IABC algorithms of interest, and state the validity and convergence conditions that they must satisfy.

Each node i maintains state v_i, with $v_i[t]$ denoting the state of node i at the *end* of the t-th iteration of the algorithm. Initial state of node i, $v_i[0]$, is equal to the initial *input* provided to node i. At the *start* of the t-th iteration ($t > 0$), the state of node i is $v_i[t-1]$. The IABC algorithms of interest will require each node i to perform the following three steps in iteration t where $t > 0$. Note that the faulty nodes may deviate from this specification.

1. *Transmit step:* Transmit current state, namely $v_i[t-1]$, on all outgoing edges and self-loop (to nodes in N_i^+ and node i itself).
2. *Receive step:* Receive values on all incoming edges and self-loop (from nodes in N_i^- and itself). Denote by $r_i[t]$ the vector of values received by node i from its incoming neighbors and itself. The size of vector $r_i[t]$ is $|N_i^-| + 1$.
3. *Update step:* Node i updates its state using a transition function Z_i as follows. Z_i is a part of the specification of the algorithm, and takes the vector $r_i[t]$ as the input.

$$v_i[t] \quad = \quad Z_i \left(r_i[t] \right) \tag{1}$$

The following conditions must be satisfied by an IABC algorithm when the set of faulty nodes (in a given execution) is F:

- *Validity:* $\forall t > 0$, and all fault-free nodes $i \in \mathcal{V} - F$,
  $$v_i[t] \geq \min_{j \in \mathcal{V}-F} v_j[t-1] \quad \text{and} \quad v_i[t] \leq \max_{j \in \mathcal{V}-F} v_j[t-1].$$[1]
- *Convergence:* for all *fault-free* nodes $i, j \in \mathcal{V} - F$, $\lim_{t \to \infty} (v_i[t] - v_j[t]) = 0$

An IABC algorithm is said to be *correct* if it satisfies the above validity and convergence conditions in the given graph $G(\mathcal{V}, \mathcal{E})$. For a given fault domain $\mathcal{F}$ for graph $G(\mathcal{V}, \mathcal{E})$, the objective here is to identify the necessary and sufficient conditions for the existence of a *correct* IABC algorithm.

4 Necessary Condition

In this section, we develop a necessary condition for the existence of a correct IABC algorithm. The necessary condition will be proved to be also sufficient in Section 6.

[1] For sets X and Y, $X - Y$ contains elements that are in X but not in Y. That is, $X - Y = \{i \mid i \in X, i \notin Y\}$.

4.1 Preliminaries

To facilitate the statement of the necessary condition, we first introduce the notions of "source component" and "reduced graph" using the following three definitions.

Definition 2 Graph Decomposition: *Let H be a directed graph. Partition graph H into strongly connected components, $H_1, H_2, \cdots, H_h$, where h is a non-zero integer dependent on graph H, such that*

- *every pair of nodes* **within** *the same strongly connected component has directed paths in H to each other, and*
- *for each pair of nodes, say i and j, that belong to two* **different** *strongly connected components, either i does not have a directed path to j in H, or j does not have a directed path to i in H.*

Construct a graph H^d wherein each strongly connected component H_k above is represented by vertex c_k, and there is an edge from vertex c_k to vertex c_l if and only if the nodes in H_k have directed paths in H to the nodes in H_l. H^d is called the decomposition graph of H.

It is known that for any directed graph H, the corresponding decomposition graph H^d is a directed *acyclic* graph (DAG) [3].

Definition 3 Source Component: *Let H be a directed graph, and let H^d be its decomposition graph as per Definition 2. Strongly connected component H_k of H is said to be a* source component *if the corresponding vertex c_k in H^d is not reachable from any other vertex in H^d.*

Definition 4 Reduced Graph: *For a given graph $G(\mathcal{V}, \mathcal{E})$ and a feasible fault set F, a reduced graph $G_F(\mathcal{V}_F, \mathcal{E}_F)$ is obtained as follows:*

- *Node set is obtained as $\mathcal{V}_F = \mathcal{V} - F$.*
- *For each node $i \in \mathcal{V}_F$, a feasible fault set $F_x(i)$ is chosen, and then the edge set $\mathcal{E}_F$ is obtained as follows:*
  - *remove from $\mathcal{E}$ all the links incident on the nodes in F, and*
  - *for each $i \in \mathcal{V}_F$ and each $j \in F_x(i) \cap \mathcal{V}_F \cap N_i^-$, remove link (j, i) from $\mathcal{E}$.*

  Feasible fault sets $F_x(i)$ and $F_x(j)$ chosen for $i \neq j$ may or may not be identical.

Note that for a given $G(\mathcal{V}, \mathcal{E})$ and a given F, multiple reduced graphs G_F may exist, depending on the choice of F_x sets above.

4.2 Necessary Condition

For a correct IABC algorithm to exist, the network graph $G(\mathcal{V}, \mathcal{E})$ must satisfy the necessary condition stated in Theorem 1 below.

Theorem 1 *Suppose that a correct IABC algorithm exists for $G(\mathcal{V}, \mathcal{E})$. Then, any reduced graph G_F, corresponding to any feasible fault set F, must contain exactly one* source component.

Proof Sketch: A complete proof is presented in Appendix A. The proof is by contradiction. Let us assume that a correct IABC algorithm exists, and for some feasible fault set F, and feasible sets $F_x(i)$ for each $i \in \mathcal{V} - F$, the resulting reduced graph contains two source components. Let L and R denote the nodes in the two source components, respectively. Thus, L and R are disjoint and non-empty. Let $C = (\mathcal{V} - F - L - R)$ be the remaining nodes in the reduced graph. C may or may not be non-empty. Assume that the nodes in F (if non-empty) are all faulty, and all the nodes in L, R, and C (if non-empty) are fault-free. Suppose that each node in L has initial input equal to m, each node in R has initial input equal to M, where $M > m$, and each node in C has an input in the range $[m, M]$. As elaborated in Appendix A, the faulty nodes can behave in such a manner that, in each iteration, nodes in L and R are forced to maintain their updated state equal to m and M, respectively, so as to satisfy the *validity* condition. This ensures that, no matter how many iterations are performed, the *convergence* condition cannot be satisfied. □

5 Algorithm 1

We will prove that there exists an IABC algorithm – particularly *Algorithm 1* below – that satisfies the *validity* and *convergence* conditions provided that the graph $G(\mathcal{V}, \mathcal{E})$ satisfies the necessary condition in Theorem 1. This implies that the necessary condition in Theorem 1 is also sufficient. *Algorithm 1* has the three-step structure described in Section 3. This algorithm is a generalization – to accommodate the generalized fault model – of iterative algorithms that were analyzed in prior work [4, 12, 7, 11], including in our own prior work as well [14, 13]. The key difference from previous algorithms is in the *Update* step below.

Algorithm 1

1. *Transmit step:* Transmit current state $v_i[t-1]$ on all outgoing edges and self-loop.

2. Receive step: Receive values on all incoming edges and self-loop. These values form vector $r_i[t]$ of size $|N_i^-|+1$ (including the value from node i itself). When a fault-free node expects to receive a message from an incoming neighbor but does not receive the message, the message value is assumed to be equal to some *default value*.

3. *Update step:* Sort the values in $r_i[t]$ in an increasing order (breaking ties arbitrarily). Let D be a vector of nodes arranged in an order "consistent" with $r_i[t]$: specifically, $D(1)$ is the node that sent the smallest value in $r_i[t]$, $D(2)$ is the node that sent the second smallest value in $r_i[t]$, and so on. The size of vector D is also $|N_i^-|+1$.

   From vector $r_i[t]$, eliminate the smallest f_1 values, and the largest f_2 values, where f_1 and f_2 are defined as follows:

   - f_1 is the largest number such that there exists a feasible fault set $F' \subseteq N_i^-$ containing nodes $D(1), D(2), ..., D(f_1)$. Recall that $i \notin N_i^-$.
   - f_2 is the largest number such that there exists a feasible fault set $F'' \subseteq N_i^-$ containing nodes $D(|N_i^-|-f_2+2), D(|N_i^-|-f_2+3), ..., D(|N_i^-|+1)$.

   F' and F'' above may or may not be identical.

   Let $N_i^*[t]$ denote the set of nodes from whom the remaining $|N_i^-|+1-f_1-f_2$ values in $r_i[t]$ were received, and let w_j denote the value received from node $j \in N_i^*[t]$. Note that $i \in N_i^*[t]$. Hence, for convenience, define $w_i = v_i[t-1]$ to be the value node i "receives" from itself. Observe that if $j \in N_i^*[t]$ is fault-free, then $w_j = v_j[t-1]$.

   Define

$$v_i[t] \;=\; Z_i(r_i[t]) \;=\; \sum_{j \in N_i^*[t]} a_i\, w_j \tag{2}$$

   where

$$a_i = \frac{1}{|N_i^*[t]|} = \frac{1}{|N_i^-|+1-f_1-f_2}$$

   The "weight" of each term on the right-hand side of (2) is a_i, and these weights add to 1. Also, $0 < a_i \leq 1$. Although f_1, f_2 and a_i may be different for each iteration t, for simplicity, we do not explicitly represent this dependence on t in the notations.

Observe $f_1 + f_2$ nodes whose values are eliminated in the *Update* step above are all in N_i^-. Thus, the above algorithm can be implemented by node i if it knows which of its incoming neighbors may fail simultaneously; node i does not need to know the entire fault domain $\mathcal{F}$ as such.

The main difference between the above algorithm and IABC algorithms in prior work is in the choice of the values eliminated from vector $r_i[t]$ in the *Update* step. The manner in which the values are eliminated ensures that the values received from nodes $D(f_1+1)$ and $D(|N_i^-|-f_2+1)$ (i.e., the smallest and largest values that survive in $r_i[t]$) are within the convex hull of the state of fault-free nodes, even if nodes $D(f_1+1)$ and $D(|N_i^-|-f_2+1)$ may not be fault-free. This property is useful in proving algorithm correctness (as discussed below).

6 Sufficiency

We will show that Algorithm 1 satisfies validity and convergence conditions, provided that $G(\mathcal{V}, \mathcal{E})$ satisfies the condition below, which matches the necessary condition stated in Theorem 1.

Sufficient condition: *Any reduced graph G_F corresponding to any feasible fault set F contains exactly one* source component.

In the rest of this section, we assume that $G(\mathcal{V}, \mathcal{F})$ satisfies the above condition. To prove its sufficiency, we first develop a *transition matrix* representation of the *Update* step in Algorithm 1.

6.1 Transition Matrix Representation

In our discussion below, $\mathbf{M}[t]$ is a square matrix, $\mathbf{M}_i[t]$ is the i-th row of the matrix, and $\mathbf{M}_{ij}[t]$ is the element at the intersection of the i-th row and j-th column of $\mathbf{M}[t]$.

For a given execution of Algorithm 1, let F denote the actual set of faulty nodes in that execution. Let $|F| = \psi$. Without loss of generality, suppose that nodes 1 through $(n - \psi)$ are fault-free, and if $\psi > 0$, nodes $(n - \psi + 1)$ through n are faulty. Denote by $v[0]$ the column vector consisting of the initial states of all the fault-free nodes. Denote by $v[t]$, where $t \geq 1$, the column vector consisting of the states of all the fault-free nodes at the end of the t-th iteration. The i-th element of vector $v[t]$ is state $v_i[t]$. The size of vector $v[t]$ is $(n - \psi)$.

We will show that the iterative update of the state of a fault-free node i $(1 \leq i \leq n-\psi)$ performed in (2) in Algorithm 1 can be expressed using the matrix form below.

$$v_i[t] = \mathbf{M}_i[t]\, v[t-1] \tag{3}$$

where $\mathbf{M}_i[t]$ is a *stochastic row* vector of size $n - \psi$. That is, $\mathbf{M}_{ij}[t] \geq 0$, for $1 \leq j \leq n - \psi$, and $\sum_{1 \leq j \leq n-\psi} \mathbf{M}_{ij}[t] = 1$.[2] By "stacking" (3) for different i, $1 \leq i \leq n - \psi$, we will represent the *Update* step of Algorithm 1 at all the fault-free nodes together using (4) below.

$$v[t] = \mathbf{M}[t]\, v[t-1] \tag{4}$$

where $\mathbf{M}[t]$ is a $(n - \psi) \times (n - \psi)$ *row stochastic* matrix, with its i-th row being equal to $\mathbf{M}_i[t]$ in (3). $\mathbf{M}[t]$ is said to be a transition matrix.

In the rest of this section, we will first "construct" a transition matrix $\mathbf{M}[t]$ that satisfies certain desirable properties. Then, we will identify a connection between the transition matrix and the sufficiency condition stated above, and use this connection to establish *convergence* property for Algorithm 1. The *validity* property also follows from the transition matrix representation.

6.2 Construction of the Transition Matrix

We will construct a transition matrix with the property described in Lemma 1 below.

Lemma 1 *The* Update step *of Algorithm 1 at the fault-free nodes can be expressed using row stochastic transition matrix $\mathbf{M}[t]$, such that there exists a feasible fault set $F_x(i)$ for each $i \in \mathcal{V} - F$ such that, for all $j \in \{i\} \cup ((\mathcal{V}_F - F_x(i)) \cap N_i^-)$,*

[2] In addition to t, the row vector $\mathbf{M}_i[t]$ may depend on the state vector $v[t-1]$ as well as the behavior of the faulty nodes in F. For simplicity, the notation $\mathbf{M}_i[t]$ does not explicitly represent this dependence.

$$\mathbf{M}_{ij}[t] \ \geq \ \beta$$

where β is a constant (to be defined later), and $0 < \beta \leq 1$.

In [13] as well, we construct a transition matrix to prove correctness of an IABC algorithm under the f-total fault model. However, the *generalized* fault model introduces additional complexity, which is handled here using a new approach to construct the transition matrix.

Proof: We prove the correctness of Lemma 1 by constructing $\mathbf{M}_i[t]$ for $1 \leq i \leq n - \psi$ that satisfies the conditions in Lemma 1. Recall that F is the set of faulty nodes, and $|F| = \psi$. As stated before, without loss of generality, nodes 1 through $n - \psi$ are assumed to be fault-free, and the remaining ψ nodes faulty.

Consider a fault-free node i performing the *Update* step in Algorithm 1. In the *Update* step, recall that the smallest f_1 and the largest f_2 values are eliminated from $r_i[t]$, where the choice of f_1 and f_2 is described in Algorithm 1. Let us denote by $\mathcal{S}$ and $\mathcal{L}$, respectively, the set of nodes[3] from whom the smallest f_1 and the largest f_2 values were received by node i in iteration t. Define sets $\mathcal{S}_g$ and $\mathcal{L}_g$ to be subsets of $\mathcal{S}$ and $\mathcal{L}$ that contain all the fault-free nodes in $\mathcal{S}$ and $\mathcal{L}$, respectively. That is, $\mathcal{S}_g = \mathcal{S} \cap (\mathcal{V} - F)$ and $\mathcal{L}_g = \mathcal{L} \cap (\mathcal{V} - F)$.

Construction of $\mathbf{M}_i[t]$ differs somewhat depending on whether sets $\mathcal{S}_g$, $\mathcal{L}_g$ and $N_i^*[t] \cap F$ are empty or non-empty. We divide the possibilities into 6 separate cases. Due to space limitation, here we present the construction for one of the cases (named Case I). The construction for the remaining cases is presented in Appendix B.

In Case I, $\mathcal{S}_g \neq \Phi$, $\mathcal{L}_g \neq \Phi$, and $N_i^*[t] \cap F \neq \Phi$. Let $m_{\mathcal{S}}$ and $m_{\mathcal{L}}$ be defined as shown below. Recall that the nodes in $\mathcal{S}_g$ and $\mathcal{L}_g$ are all fault-free, and therefore, for any node $j \in \mathcal{S}_g \cup \mathcal{L}_g$, $w_j = v_j[t-1]$ (in the notation of Algorithm 1).

$$m_{\mathcal{S}} = \frac{\sum_{j \in \mathcal{S}_g} v_j[t-1]}{|\mathcal{S}_g|} \quad \text{and} \quad m_{\mathcal{L}} = \frac{\sum_{j \in \mathcal{L}_g} v_j[t-1]}{|\mathcal{L}_g|}$$

Now, consider any node $k \in N_i^*[t]$. By the definition of sets $\mathcal{S}_g$ and $\mathcal{L}_g$, $m_{\mathcal{S}} \leq w_k \leq m_{\mathcal{L}}$. Therefore, we can find weights $S_k \geq 0$ and $L_k \geq 0$ such that $S_k + L_k = 1$, and

$$\begin{aligned} w_k &= S_k \, m_{\mathcal{S}} + L_k \, m_{\mathcal{L}} && (5) \\ &= \frac{S_k}{|\mathcal{S}_g|} \sum_{j \in \mathcal{S}_g} v_j[t-1] + \frac{L_k}{|\mathcal{L}_g|} \sum_{j \in \mathcal{L}_g} v_j[t-1] && (6) \end{aligned}$$

Clearly, at least one of S_k and L_k must be $\geq 1/2$. We now define elements $\mathbf{M}_{ij}[t]$ of row $\mathbf{M}_i[t]$:

- For $j \in N_i^*[t] \cap (\mathcal{V} - F)$: In this case, j is either a fault-free incoming neighbor of i, or i itself. For each such j, define $\mathbf{M}_{ij}[t] = a_i$. This is obtained by observing in (2) that the contribution of such a node j to the new state $v_i[t]$ is $a_i \, w_j = a_i \, v_j[t-1]$.

  The elements of $\mathbf{M}_i[t]$ defined here add up to

$$|N_i^*[t] \cap (\mathcal{V} - F)| \, a_i$$

[3] Although $\mathcal{S}$ and $\mathcal{L}$ may be different for each t, for simplicity, we do not explicitly represent this dependence on t in the notations $\mathcal{S}$ and $\mathcal{L}$.

- For $j \in \mathcal{S}_g \cup \mathcal{L}_g$: In this case, j is a fault-free node in $\mathcal{S}$ or $\mathcal{L}$.

  For each $j \in \mathcal{S}_g$,

$$\mathbf{M}_{ij}[t] \;=\; a_i \sum_{k \in N_i^*[t] \cap F} \frac{S_k}{|\mathcal{S}_g|}$$

  and for each node $j \in \mathcal{L}_g$,

$$\mathbf{M}_{ij}[t] \;=\; a_i \sum_{k \in N_i^*[t] \cap F} \frac{L_k}{|\mathcal{L}_g|}$$

  To obtain these two expressions, we represent value w_k sent by each faulty node k in $N_i^*[t]$, i.e., $k \in N_i^*[t] \cap F$, using (6). Recall that this node k contributes $a_i w_k$ to (2). The above two expressions are then obtained by summing (6) over all the faulty nodes in $N_i^*[t] \cap F$, and replacing this sum by equivalent contributions by nodes in $\mathcal{S}_g$ and $\mathcal{L}_g$.

  The elements of $\mathbf{M}_i[t]$ defined here add up to

$$a_i \sum_{k \in N_i^*[t] \cap F} (S_k + L_k) = |N_i^*[t] \cap F|\, a_i.$$

- For $j \in (\mathcal{V} - F) - (N_i^*[t] \cup \mathcal{S}_g \cup \mathcal{L}_g)$: These fault-free nodes have not yet been considered above. For each such node j, define $\mathbf{M}_{ij}[t] = 0$.

With the above definition of $\mathbf{M}_i[t]$, it should be easy to see that $\mathbf{M}_i[t]\, v[t-1]$ is, in fact, identical to $v_i[t]$ obtained using (2). Thus, the above construction of $\mathbf{M}_i[t]$ results in the contribution of the faulty nodes in $N_i^*[t]$ to (2) being replaced by an equivalent contribution from fault-free nodes in $\mathcal{L}_g$ and $\mathcal{S}_g$.

Properties of $\mathbf{M}_i[t]$: First, we show that $\mathbf{M}[t]$ is row stochastic. Observe that all the elements of $\mathbf{M}_i[t]$ are non-negative. Also, all the elements of $\mathbf{M}_i[t]$ above add up to

$$|N_i^*[t] \cap (\mathcal{V} - F)|\, a_i + |N_i^*[t] \cap F|\, a_i = |N_i^*[t]|\, a_i = 1$$

because $a_i = 1/|N_i^*[t]|$ as defined in Algorithm 1. Thus, $\mathbf{M}_i[t]$ is a stochastic row vector.

Recall that from the above discussion, for $k \in N_i^*[t]$, one of S_k and L_k must be $\geq 1/2$. Without loss of generality, assume that $S_s \geq 1/2$ for some $s \in N_i^*[t] \cap F$. Consequently, for each node $j \in \mathcal{S}_g$, $\mathbf{M}_{ij}[t] \geq \frac{a_i}{|\mathcal{S}_g|} S_s \geq \frac{a_i}{2|\mathcal{S}_g|}$. Also, for each fault-free node j in $N_i^*[t]$, $\mathbf{M}_{ij}[t] = a_i$. Thus, if β is chosen such that

$$0 < \beta \leq \frac{a_i}{2|\mathcal{S}_g|} \tag{7}$$

and $F_x(i)$ is defined to be equal to $\mathcal{L}$, then the condition in the lemma holds for node i. That is, $\mathbf{M}_{ij}[t] \geq \beta$ for $j \in \{i\} \cup ((\mathcal{V}_F - F_x(i)) \cap N_i^-)$.

All Cases Together: Using similar constructions in other cases as well (presented in Appendix B) and a suitable choice of β (presented in Appendix C), we can obtain a row stochastic matrix $\mathbf{M}[t]$, and for each $i \in \mathcal{V} - F$ identify a feasible fault set $F_x(i)$, such that $\mathbf{M}_{ij}[t] \geq \beta$ for all $j \in \{i\} \cup ((\mathcal{V}_F - F_x(i)) \cap N_i^-)$. Thus, Lemma 1 can be proved correct.

□

6.3 Validity and Convergence of Algorithm 1

The rest of the proof structure is derived from our previous work wherein we proved the correctness of an IABC algorithm for the f-total fault model [13]. Let R_F denote the set of all the reduced graphs of $G(\mathcal{V}, \mathcal{E})$ corresponding to a feasible fault set F. Let $\tau = |R_F|$. τ depends on F and the underlying network, and is finite.

In this discussion, let us denote a reduced graph by an italic upper case letter, and the corresponding "connectivity matrix" (defined below) using the same letter in boldface upper case. Thus, $\mathbf{H}$ denotes the connectivity matrix for graph $H \in R_F$.

Non-zero elements of connectivity matrix $\mathbf{H}$ are defined as follows: (i) for $1 \leq i, j \leq n - \psi$, $\mathbf{H}_{ij} = 1$ if and only if $(j, i) \in H$, and (ii) $\mathbf{H}_{ii} = 1$ for $1 \leq i \leq n - \psi$. That is, non-zero elements of row $\mathbf{H}_i$ correspond to the incoming links at node i, and the self-loop at node i. Thus, the connectivity matrix for any reduced graph in R_F has a non-zero diagonal.

Based on the *sufficient condition* stated at the start of Section 6 and Lemma 1, we can show the following key lemmas. The proofs are presented in Appendix D and E.

Lemma 2 *For any $H \in R_F$, $\mathbf{H}^{\mathbf{n}-\psi}$ has at least one non-zero column.*

Lemma 3 *For any $t \geq 1$, there exists a graph $H \in R_F$ such that $\beta\mathbf{H} \leq \mathbf{M}[t]$.*

Theorem 2 *Suppose that $G(\mathcal{V}, \mathcal{E})$ satisfies the sufficient condition stated above. Algorithm 1 satisfies both the validity and convergence conditions.*

Proof: A complete proof is presented in Appendix F. By repeated application of (4), we can represent the *Update* step of Algorithm 1 at the t-th iterations ($t \geq 1$) as:

$$v[t] \quad = \quad \left(\Pi_{i=1}^{t} \mathbf{M}[i]\right) v[0] \tag{8}$$

where $\mathbf{M}[i]$ is constructed as described above. When presenting matrix products, for convenience of presentation, we adopt the following convention: for $a < b$, $\Pi_{i=a}^{b} \mathbf{A}[i]$ denotes the "backward" product $\mathbf{A}[b]\mathbf{A}[b-1]\cdots\mathbf{A}[a]$. Thus, $\Pi_{i=1}^{t} \mathbf{M}[i]$ in (8) above represents $\mathbf{M}[t]\mathbf{M}[t-1]\cdots\mathbf{M}[1]$.

Since $\mathbf{M}[i]$ is row stochastic, then from (4), it follows that Algorithm 1 satisfies the validity condition. Based on Lemmas 2 and 3, we can also show that the rows of $\Pi_{i=1}^{t} \mathbf{M}[i]$ become identical in the limit (as elaborated in Appendix F). This observation and (8) together imply that the states of the fault-free nodes satisfy the convergence condition too. □

7 Conclusions

This paper considers a *generalized* fault model, which can be used to specify more complex failure patterns, such as correlated failures or non-uniform node reliabilities. Under this fault model, we prove a *tight* necessary and sufficient condition for the existence of synchronous iterative approximate Byzantine consensus algorithms in arbitrary directed graphs. The analysis of consensus under the generalized fault model sheds new light on how the fault model affects algorithm design.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Optimization and Neural Computation Series. Athena Scientific, 1997.

[2] V. Bhandari and N. H. Vaidya. On reliable broadcast in a radio network. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, PODC '05, pages 138–147, New York, NY, USA, 2005. ACM.

[3] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Higher Education, 2006.

[4] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl. Reaching approximate agreement in the presence of faults. *J. ACM*, 33:499–516, May 1986.

[5] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32:374–382, April 1985.

[6] J. Hajnal. Weak ergodicity in non-homogeneous markov chains. In *Proceedings of the Cambridge Philosophical Society*, volume 54, pages 233–246, 1958.

[7] R. M. Kieckhafer and M. H. Azadmanesh. Low cost approximate agreement in partially connected networks. *Journal of Computing and Information*, 3(1):53–85, 1993.

[8] C.-Y. Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, PODC '04, pages 275–282, New York, NY, USA, 2004. ACM.

[9] P. Kuznetsov. Understanding non-uniform failure models. *Bulletin of the European Association for Theoretical Computer Science (BEATCS)*, 106:53–77, 2012.

[10] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. on Programming Languages and Systems*, 1982.

[11] H. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos. Consensus of multi-agent networks in the presence of adversaries using only local information. *HiCoNs*, 2012.

[12] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[13] N. H. Vaidya. Matrix representation of iterative approximate byzantine consensus in directed graphs. *CoRR*, Mar. 2012.

[14] N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. volume abs/1201.4183, 2012.

[15] J. Wolfowitz. Products of indecomposable, aperiodic, stochastic matrices. In *Proceedings of the American Mathematical Society*, volume 14, pages 733–737, 1963.

[16] H. Zhang and S. Sundaram. Robustness of information diffusion algorithms to locally bounded adversaries. *CoRR*, abs/1110.3843, 2011.

APPENDIX

A Necessity Proof in Section 4

Now, we present the proof for Theorem 1. The proof is by contradiction. Let us assume that a correct IABC algorithm exists, and for some feasible fault set F, and feasible sets $F_x(i)$ for each $i \in \mathcal{V} - F$, the resulting reduced graph contains two source components.

Let L and R denote the nodes in the two source components, respectively. Thus, L and R are disjoint and non-empty. Let $C = (\mathcal{V} - F - L - R)$ be the remaining nodes in the reduced graph. C may or may not be non-empty. Let us now assume that the nodes in F (if non-empty) are all faulty, and all the nodes in L, R, and C (if non-empty) are fault-free.

Consider the case when (i) each node in L has initial input m , (ii) each node in R has initial input M, such that $M > m$, and (iii) each node in C (if non-empty) has an input in the interval $[m, M]$.

In the *Transmit step* of iteration 1 of the IABC algorithm, suppose that the faulty nodes in F (if non-empty) send $m^- < m$ on outgoing links to nodes in L, send $M^+ > M$ on outgoing links to nodes in R, and send some arbitrary value in interval $[m, M]$ on outgoing links to nodes in C (if non-empty). This behavior is possible since nodes in F are Byzantine faulty. Note that $m^- < m < M < M^+$. Each fault-free node $k \in \mathcal{V} - F$ sends to nodes in N_k^+ value $v_k[0]$ in iteration 1.

Consider any node $i \in L$. Since L is a source component in the reduced graph, it must be true that $N_i^- \cap (C \cup R) \subseteq N_i^- \cap F_x(i) \cap \mathcal{V}_F$.[4]

Now, node i receives m^- from the nodes in $N_i^- \cap F$, and values in $[m, M]$ from the nodes in $N_i^- \cap (C \cup R)$, and m from the nodes in $\{i\} \cup (N_i^- \cap L)$. Figure 1 illustrates the behavior of faulty nodes in F and the value received by node i.

Consider the following two cases:

- $N_i^- \cap F$ **and** $N_i^- \cap (C \cup R)$ **are both non-empty:** In this case, $(N_i^- \cap F) \subseteq F$ and $N_i^- \cap (C \cup R) = N_i^- \cap F_x(i) \cap \mathcal{V}_F \subseteq F_x(i)$. From node i's perspective, consider two possible scenarios: (a) nodes in $N_i^- \cap F$ are all faulty, and the other nodes are fault-free, and (b) nodes in $N_i^- \cap (C \cup R) = N_i^- \cap F_x(i) \cap \mathcal{V}_F$ are all faulty, and the other nodes are fault-free. Note that, since $F_x(i)$ is a feasible fault set, $N_i^- \cap F_x(i) \cap \mathcal{V}_F$ is also a feasible fault set. Similarly, since F is a feasible fault set, $N_i^- \cap F$ is also a feasible fault set.

  In scenario (a), from node i's perspective, the fault-free nodes have sent values in interval $[m, M]$, whereas the faulty incoming neighbors, i.e., nodes in $N_i^- \cap F$, have sent value m^-. According to the validity condition, $v_i[1] \geq m$. On the other hand, in scenario (b), the fault-free incoming neighbors have sent values m^- and m, where $m^- < m$; so $v_i[1] \leq m$, according to the validity condition. Since node i does not know whether the correct scenario is (a) or (b), it must update its state to satisfy the validity condition in both cases. Thus, it follows that $v_i[1] = m$.

[4]Explanation: In the reduced graph, there are no incoming links at i from nodes in $N_i^- \cap (C \cup R)$. Thus, any incoming links in $\mathcal{E}$ from the nodes in $N_i^- \cap (C \cup R)$ must have been removed when constructing $\mathcal{E}_F$ for the reduced graph. Recall that when constructing $\mathcal{E}_F$, incoming links from nodes in $N_i^- \cap F_x(i) \cap \mathcal{V}_F$ are removed. It should be noted that the algorithm is performed using the links in $\mathcal{E}$, not the reduced graph. Thus, in the *Transmit step*, all links in $\mathcal{E}$ are used.

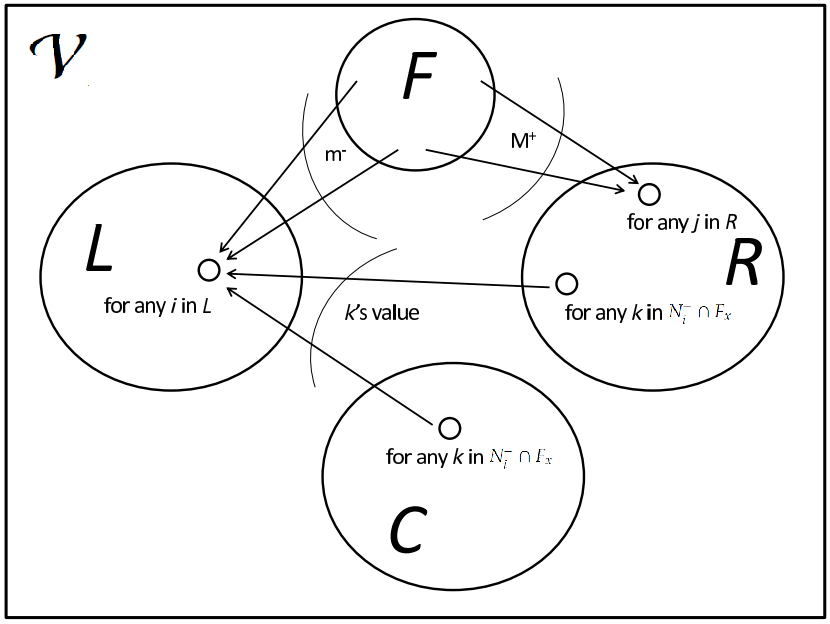

Figure 1: Illustration of the behavior of faulty nodes in F and the value received at node i.

- **At most one of $N_i^- \cap F$ and $N_i^- \cap (C \cup R)$ is non-empty:** Recall that $N_i^- \cap F$ and $N_i^- \cap (C \cup R) = N_i^- \cap F_x(i) \cap \mathcal{V}_F$ are both feasible fault sets. Since at least one of these two sets is empty, their union, i.e., $(N_i^- \cap F) \cup (N_i^- \cap (C \cup R))$, is also a feasible fault set.

  Then, from node i's perspective, it is possible that all the nodes in $(N_i^- \cap F) \cup (N_i^- \cap (C \cup R))$ are faulty, and the rest of the nodes are fault-free. In this situation, the values sent to node i by the fault-free nodes (which are all in $\{i\} \cup (N_i^- \cap L)$) are all m, and therefore, $v_i[1]$ must be set to m as per the validity condition.

Hence, $v_i[1] = m$ for each node $i \in L$. Similarly, we can show that $v_j[1] = M$ for each node $j \in R$.

Now consider the nodes in set C (if non-empty). All the values received by the nodes in C are in $[m, M]$, therefore, their new state must also remain in $[m, M]$, as per the validity condition.

The above discussion implies that, at the end of iteration 1, the following conditions hold true: (i) state of each node in L is m , (ii) state of each node in R is M, and (iii) state of each node in C (if non-empty) is in the interval $[m, M]$. These conditions are identical to the initial conditions listed previously. Then, by a repeated application of the above argument (proof by induction), it follows that for any $t \geq 0$, $v_i[t] = m$ for all nodes $i \in L$, $v_j[t] = M$ for all nodes $j \in R$ and $v_k[t] \in [m, M]$ for all nodes $k \in C$.

Since L and R both contain fault-free nodes, and $m \neq M$, the *convergence* requirement is not satisfied. This is a contradiction to the assumption that a correct iterative algorithm exists in $G(\mathcal{V}, \mathcal{E})$.

B Construction for other Cases in Section 6.2

When discussing Case I in Section 6.2, we deferred discussion of the other cases. We present the construction for the rest of the cases here. There are six cases in total:

- Case I: $\mathcal{S}_g \neq \Phi$, $\mathcal{L}_g \neq \Phi$, and $N_i^*[t] \cap F \neq \Phi$.
- Case II: $\mathcal{S}_g \neq \Phi$, $\mathcal{L}_g \neq \Phi$, and $N_i^*[t] \cap F = \Phi$.
- Case III: $\mathcal{S}_g = \Phi$, $\mathcal{L}_g \neq \Phi$, and $N_i^*[t] \cap F \neq \Phi$.
- Case IV: $\mathcal{S}_g \neq \Phi$, $\mathcal{L}_g = \Phi$, and $N_i^*[t] \cap F \neq \Phi$.
- Case V: $\mathcal{S}_g = \Phi$, $\mathcal{L}_g = \Phi$, and $N_i^*[t] \cap F \neq \Phi$.
- Case VI: at most one of $\mathcal{S}_g$ and $\mathcal{L}_g$ is non-empty, and $N_i^*[t] \cap F = \Phi$.

Note that the choice of f_1 and f_2 in Algorithm 1 ensures that the value from node i itself is never dropped from $r_i[t]$; therefore, $i \in N_i^*[t]$, and $N_i^*[t]$ is always non-empty.

B.1 Case II

Now, we consider the case when $\mathcal{S}_g \neq \Phi$, $\mathcal{L}_g \neq \Phi$, and $N_i^*[t] \cap F = \Phi$. That is, when each of $\mathcal{S}$ and $\mathcal{L}$ contains at least one fault-free node, and $N_i^*[t]$ contains only fault-free node(s). In fact, the analysis of Case II is very similar to the analysis presented in Section 6.2 for Case I when $N_i^*[t]$ does contain a faulty node.

We now discuss how the analysis of Case I can be applied to Case II. Rewrite (2) as follows:

$$v_i[t] = \frac{a_i}{2} v_i[t-1] + \frac{a_i}{2} v_i[t-1] + \sum_{j \in N_i^*[t]-\{i\}} a_i w_j \tag{9}$$

$$= a_i w_z + a_i w_i + \sum_{j \in N_i^*[t]-\{i\}} a_i w_j \tag{10}$$

In the above equation, z is to be viewed as a "virtual" incoming neighbor of node i, which has sent value $w_z = \frac{v_i[t-1]}{2}$ to node i in iteration t. With the above rewriting of state update, the value received by node i from itself should be viewed as $w_i = \frac{v_i[t-1]}{2}$ instead of $v_i[t-1]$. With this transformation, Case II now becomes identical to Case I, with virtual node z being treated as an incoming neighbor of node i.

In essence, a part of node i's contribution (half, to be precise) is now replaced by equivalent contribution by nodes in $\mathcal{L}_g$ and $\mathcal{S}_g$. We now define elements $\mathbf{M}_{ij}[t]$ of row $\mathbf{M}_i[t]$:

- For $j = i$: $\mathbf{M}_{ij}[t] = \frac{a_i}{2}$. This is obtained by observing in (2) that node i's contribution to the new state $v_i[t]$ is $a_i \frac{v_i[t-1]}{2}$.

- For $j \in N_i^*[t] - \{i\}$: In this case, j is a fault-free incoming neighbor of i. For each such j, define $\mathbf{M}_{ij}[t] = a_i$. This is obtained by observing in (2) that the contribution of node j to the new state $v_i[t]$ is $a_i w_j = a_i v_j[t-1]$.

- For $j \in \mathcal{S}_g \cup \mathcal{L}_g$: In this case, j is a fault-free node in $\mathcal{S}$ or $\mathcal{L}$.

  For each $j \in \mathcal{S}_g$,

$$\mathbf{M}_{ij}[t] = \frac{a_i}{2} \frac{S_z}{|\mathcal{S}_g|}$$

  and for each node $j \in \mathcal{L}_g$,

$$\mathbf{M}_{ij}[t] = \frac{a_i}{2} \frac{L_z}{|\mathcal{L}_g|}$$

  where S_z and L_z are chosen such that $S_z + L_z = 1$ and $w_z = \frac{v_i[t-1]}{2} = \frac{S_z}{2} m_{\mathcal{S}} + \frac{L_z}{2} m_{\mathcal{L}}$. Note that such S_z and L_z exist because by definition of $\mathcal{S}_g$ and $\mathcal{L}_g$, $v_i[t-1] \geq w_j$, $\forall j \in S_g$ and $v_i[t-1] \leq w_j$, $\forall j \in L_g$. Then the two expressions above are obtained by replacing the contribution of the virtual node z by an equivalent contribution by the nodes in $\mathcal{S}_g$ and $\mathcal{L}_g$, respectively.

- For $j \in (\mathcal{V} - F) - (N_i^*[t] \cup \mathcal{S}_g \cup \mathcal{L}_g)$: These fault-free nodes have not yet been considered above. For each such node j, define $\mathbf{M}_{ij}[t] = 0$.

By argument similar to that in Section 6.2, $\mathbf{M}[t]$ is row stochastic. Without loss of generality, suppose that $S_z \geq 1/2$. Then for each node $j \in \mathcal{S}_g$, $\mathbf{M}_{ij}[t] = \frac{a_i}{2|\mathcal{S}_g|} S_z \geq \frac{a_i}{4|\mathcal{S}_g|}$. Also, for fault-free node j in $N_i^*[t] - \{i\}$, $\mathbf{M}_{ij}[t] = a_i$, and $\mathbf{M}_{ii}[t] = \frac{a_i}{2}$. Recall that by definition, $|\mathcal{S}_g| \geq 1$. Hence, if β is chosen such that

$$0 < \beta \leq \frac{a_i}{4|\mathcal{S}_g|} \tag{11}$$

and $F_x(i)$ is defined to be equal to $\mathcal{L}$, then the condition in the Lemma 1 holds for node i. That is, $\mathbf{M}_{ij}[t] \geq \beta$ for $j \in \{i\} \cup (\mathcal{V}_F - F_x(i)) \cap N_i^-$.

B.2 Cases III and IV

Now, we describe the construction of Case III. The construction for Case IV is very similar, and thus, is omitted here.

In Case III, $\mathcal{S}_g = \Phi, \mathcal{L}_g \neq \Phi$, and $N_i^*[t] \cap F \neq \Phi$. Thus, $\mathcal{S}$ does not contain any fault-free nodes (hence $\mathcal{S}_g$ is empty). This may be due to one of the following two reasons: (i) the set $\mathcal{S}$ is non-empty, but all the nodes in $\mathcal{S}$ are faulty, or (ii) set $\mathcal{S}$ is empty.

Assume that $l \in \mathcal{L}$ is a fault-free node, and that all the nodes in $\mathcal{S}$ are faulty (i.e., $\mathcal{S}_g = \Phi$) or that $\mathcal{S}$ is empty (i.e., $f_1 = 0$). In this case, observe that node $D(f_1 + 1)$ must be fault-free (otherwise, f_1 cannot be the largest value as defined in Algorithm 1). Now, consider any node $k \in N_i^*[t]$. Similar to the argument in Case I, we can find weights $S_k \geq 0$ and $L_k \geq 0$ such that

$$S_k + L_k = 1$$

and

$$w_k = S_k\, v_{D(f_1+1)}[t-1] + L_k\, v_l[t-1] \tag{12}$$

We now define $\mathbf{M}_{ij}[t]$ for all fault-free j.

- For $j \in (N_i^*[t] - \{D(f_1+1)\}) \cap (\mathcal{V} - F)$. That is, j is a fault-free node in $N_i^*[t]$ with the exception of $D(f_1+1)$.

  For each such j, define $\mathbf{M}_{ij}[t] = a_i$. This is obtained by observing in (2) that the contribution of node j to the new state $v_i[t]$ is $a_i w_j = a_i\, v_j[t-1]$.

  The elements of $\mathbf{M}_i[t]$ defined here (including the case of $j = i$) add up to

  $$(|N_i^*[t] \cap (\mathcal{V} - F)| - 1)\, a_i.$$

- For nodes $D(f_1+1)$ and l: Define

  $$\mathbf{M}_{iD(f_1+1)}[t] = a_i + \sum_{k \in N_i^*[t] \cap F} a_i\, S_k$$

  and

  $$\mathbf{M}_{il}[t] = \sum_{k \in N_i^*[t] \cap F} a_i\, L_k$$

  Similar to Case I presented in Section 6.2, these two expressions are obtained by summing up the contribution over the faulty nodes in $N_i^*[t]$, and replacing the sum by an equivalent contribution by the nodes $D(f_1+1)$ and l, respectively, according to (12).

  The above elements of $\mathbf{M}_i[t]$ add up to

  $$a_i \left(1 + \sum_{k \in N_i^*[t] \cap F} (S_k + L_k)\right) = (1 + |N_i^*[t] \cap F|)\, a_i.$$

- For $j \in (\mathcal{V} - F) - (N_i^*[t] \cup \{l\})$: These fault-free nodes have not yet been considered above. For each such j, define $M_{ij}[t] = 0$.

Similar to Case I, in Case III as well, it should be easy to see that

$$\mathbf{M}_i[t]\, v[t-1]$$

is identical to $v_i[t]$ obtained using (2).

Properties of $\mathbf{M}_i[t]$: All the elements of $\mathbf{M}_i[t]$ are non-negative. The elements of $\mathbf{M}_i[t]$ defined in Case II add up to

$$(|N_i^*[t] \cap (\mathcal{V} - F)| - 1)\, a_i + (1 + |N_i^*[t] \cap F|)\, a_i = |N_i^*[t]|\, a_i = 1$$

Thus, $\mathbf{M}_i[t]$ is a stochastic row vector.

In Case III, recall that for any fault-free node j in $N_i^*[t]$ (including $j = D(f_1+1)$ and $j = i$), $\mathbf{M}_{ij}[t] \geq a_i$. Thus, if β is chosen such that

$$0 < \beta \leq a_i \tag{13}$$

and $F_x(i)$ is defined to be equal to $\mathcal{L}$, then the condition in the Lemma 1 holds for node i.

B.3 Case V

Consider Case V, where $N_i^*[t] \cap F \neq \Phi$, and $\mathcal{S}_g = \mathcal{L}_g = \Phi$. In this case, it should be easy to see that $N_i^*[t]$ contains at least 3 nodes. In particular, D_{f_1+1} must be fault-free (otherwise, f_1 cannot be maximum possible), $D_{|N_i^-|-f_2+1}$ must be fault-free (otherwise, f_2 cannot be maximum possible), and there is a faulty node in $N_i^*[t]$.

Now this case can be handled similar to Case III analyzed above. In particular, entries in $\mathbf{M}_i[t]$ are defined similarly with l being defined equal to $D_{N_i^- - f_2+1}$. Also, define $F_x(i) = \Phi$.

Hence, it is easy to see that the properties of $\mathbf{M}_i[t]$ are identical to Case III presented above.

B.4 Case VI

Here, we consider the case when at most one of $\mathcal{S}$ and $\mathcal{L}$ contains a fault-free node and $N_i^*[t] \cap F = \Phi$. Without loss of generality, suppose that $\mathcal{S}$ contains only faulty nodes, and $\mathcal{L}$ may contain a fault-free node.

In this case, define $\mathbf{M}_{ij}[t] = a_i$ for $j \in N_i^*[t]$; define $\mathbf{M}_{ij} = 0$ for all other fault-free nodes j. Also, define $F_x(i) = \mathcal{L}$.

The properties of $\mathbf{M}_i[t]$ thus defined are identical to Case III above.

C Putting Cases Together

Now, let us consider Cases I-VI together. From the definition of a_i in Algorithm 1, observe that $a_i \geq \frac{1}{|N_i^-|+1}$ (because $f_1, f_2 \geq 0$). Let us define

$$\alpha = \min_{i \in \mathcal{V}} \frac{1}{|N_i^-| + 1}$$

Moreover, observe that $|\mathcal{S}_g| \leq n$ and $|\mathcal{L}_g| \leq n$. Then define β as

$$\beta = \frac{\alpha}{4n} \tag{14}$$

This definition satisfies constraints on β in Cases I through VI (conditions (7), (11) and (13)). Thus, Lemma 1 holds for all six cases with this choice of β.

D Proof of Lemma 2 in Section 6.3

Here, we present the proof of the first key lemma used in the sufficiency proof.

Lemma 2 *For any $H \in R_F$, $\mathbf{H}^{\mathbf{n}-\psi}$ has at least one non-zero column.*

Proof: $G(\mathcal{V}, \mathcal{E})$ satisfies the *sufficient condition* stated at the start of Section 6. Therefore, there exists at least one non-faulty node k in the reduced graph H that has directed paths to all the nodes in H (consisting of the edges in H). Since the length of the path from k to any other node in H is at most $n - \psi - 1$, the k-th column of matrix $\mathbf{H}^{n-\psi}$ will be non-zero.[5] □

E Proof of Lemma 3 in Section 6.3

Here, we present the proof of the second key lemma used in the sufficiency proof. We start with two definitions:

Definition 5 *For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of identical size, and a scalar γ, $\gamma \boldsymbol{B} \leq \boldsymbol{A}$ provided that $\gamma \boldsymbol{B}_{ij} \leq \boldsymbol{A}_{ij}$ for all i, j.*

We want to prove the following lemma.

Lemma 3 *For any $t \geq 1$, there exists a graph $H \in R_F$ such that $\beta \mathbf{H} \leq \mathbf{M}[t]$.*

Proof: Observe that the i-th row of the transition matrix $\mathbf{M}[t]$ corresponds to the state update (in Algorithm 1) performed at fault-free node i. Recall from Lemma 1 that $\mathbf{M}_{ij}[t] \geq \beta$ for $j \in \{i\} \cup ((\mathcal{V}_F - F_x(i)) \cap N_i^-)$, where $F_x(i)$ is a feasible fault set.

Let us obtain a reduced graph H by choosing $F_x(i)$ for each i as defined in Lemma 1. Then from the definition of connectivity matrix $\mathbf{H}$, Lemma 3 then follows. □

F Correctness of Algorithm 1

When presenting matrix products, for convenience of presentation, we adopt the following convention: for $a < b$, $\Pi_{i=a}^{b} \mathbf{A}[i]$ denotes the "backward" product $\mathbf{A}[b]\mathbf{A}[b-1] \cdots \mathbf{A}[a]$.

The proof below is similar to a proof for the f-total fault model in our previous work [13]. It is included here for the convenience of the referees.

F.1 Matrix Preliminaries

In the discussion below, we use boldface upper case letters to denote matrices, rows of matrices, and their elements. For instance, $\mathbf{H}$ denotes a matrix, $\mathbf{H}_i$ denotes the i-th row of matrix $\mathbf{H}$, and $\mathbf{H}_{ij}$ denotes the element at the intersection of the i-th row and the j-th column of matrix $\mathbf{H}$.

Definition 6 *A vector is said to be* stochastic *if all the elements of the vector are* non-negative*, and the elements add up to 1. A matrix is said to be row stochastic if each row of the matrix is a stochastic vector.*

[5] That is, all the elements of the column will be non-zero. Also, such a non-zero column will exist in $\mathbf{H}^{n-\psi-1}$, too. We use the loose bound of $n - \psi$ to simplify the presentation.

For a row stochastic matrix $\mathbf{A}$, coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as follows [15]:

$$\begin{aligned}\delta(\mathbf{A}) &= \max_j \max_{i_1,i_2} |\mathbf{A}_{i_1 j} - \mathbf{A}_{i_2 j}| \\ \lambda(\mathbf{A}) &= 1 - \min_{i_1,i_2} \sum_j \min(\mathbf{A}_{i_1 j}, \mathbf{A}_{i_2 j})\end{aligned}$$

It is easy to show that $0 \leq \delta(\mathbf{A}) \leq 1$ and $0 \leq \lambda(\mathbf{A}) \leq 1$, and that the rows of $\mathbf{A}$ are all identical if and only if $\delta(\mathbf{A}) = 0$. Also, $\lambda(\mathbf{A}) = 0$ if and only if $\delta(\mathbf{A}) = 0$.

The next result from [6] establishes a relation between the coefficient of ergodicity $\delta(\cdot)$ of a product of row stochastic matrices, and the coefficients of ergodicity $\lambda(\cdot)$ of the individual matrices defining the product.

Lemma 4 *For any p square row stochastic matrices $\mathbf{Q}(1), \mathbf{Q}(2), \ldots \mathbf{Q}(p)$,*

$$\delta(\mathbf{Q}(p)\mathbf{Q}(p-1)\cdots\mathbf{Q}(1)) \leq \Pi_{i=1}^{p} \lambda(\mathbf{Q}(i)).$$

Lemma 4 is proved in [6]. It implies that if, for all i, $\lambda(\mathbf{Q}(i)) \leq 1-\gamma$ for some γ, where $0 < \gamma \leq 1$, then $\delta(\mathbf{Q}(p)\mathbf{Q}(p-1)\cdots\mathbf{Q}(1))$ will approach zero as p approaches ∞. We now define a *scrambling* matrix [6, 15].

Definition 7 *A row stochastic matrix $\mathbf{H}$ is said to be a* scrambling *matrix if $\lambda(\mathbf{H}) < 1$.*

The following lemma follows easily from the above definition of $\lambda(\cdot)$.

Lemma 5 *If any column of a row stochastic matrix $\mathbf{H}$ contains only non-zero elements that are all lower bounded by some constant γ, where $0 < \gamma \leq 1$, then $\mathbf{H}$ is a scrambling matrix, and $\lambda(\mathbf{H}) \leq 1-\gamma$.*

F.2 Correctness of Algorithm 1

Lemma 6 *For any $z \geq 1$, in the product below of $\mathbf{H}[t]$ matrices for consecutive $\tau(n-\psi)$ iterations, at least one column is non-zero.*

$$\Pi_{t=z}^{z+\tau(n-\psi)-1} \mathbf{H}[t]$$

Proof: Since the above product consists of $\tau(n-\psi)$ connectivity matrices corresponding to graphs in $R_{\mathcal{F}}$, at least one of the connectivity matrices corresponding to the τ distinct graphs in $R_{\mathcal{F}}$, say matrix $\mathbf{H}_*$, will appear in the above product at least $n-\psi$ times.

Now observe that: (i) By Lemma 2, $\mathbf{H}_*^{n-\psi}$ contains a non-zero column, say the k-th column is non-zero, and (ii) all the $\mathbf{H}[t]$ matrices in the product contain a non-zero diagonal. These two observations together imply that the k-th column in the above product is non-zero. □

Let us now define a sequence of matrices $\mathbf{Q}(i)$, $i \geq 1$, such that each of these matrices is a product of $\tau(n-\psi)$ of the $\mathbf{M}[t]$ matrices. Specifically,

$$\mathbf{Q}(i) = \Pi_{t=(i-1)\tau(n-\psi)+1}^{i\tau(n-\psi)} \mathbf{M}[t] \tag{15}$$

From (8) and (15) observe that

$$v[k\tau(n-\psi)] \;=\; \left(\Pi_{i=1}^{k}\,\mathbf{Q}(i)\right)\,v[0] \tag{16}$$

Lemma 7 *For $i \geq 1$, $\mathbf{Q}(i)$ is a scrambling row stochastic matrix, and*

$$\lambda(\mathbf{Q}(i)) \leq 1-\beta^{\tau(n-\psi)}.$$

Proof: $\mathbf{Q}(i)$ is a product of row stochastic matrices ($\mathbf{M}[t]$); therefore, $\mathbf{Q}(i)$ is row stochastic. From Lemma 3, for each $t \geq 1$,

$$\beta\,\mathbf{H}[t] \;\leq\; \mathbf{M}[t]$$

Therefore,

$$\beta^{\tau(n-\psi)}\,\Pi_{t=(i-1)\tau(n-\psi)+1}^{i\tau(n-\psi)}\,\mathbf{H}[t] \;\leq\; \Pi_{t=(i-1)\tau(n-\psi)+1}^{i\tau(n-\psi)}\,\mathbf{M}[t] \;=\; \mathbf{Q}(i)$$

By using $z = (i-1)(n-\psi)+1$ in Lemma 6, we conclude that the matrix product on the left side of the above inequality contains a non-zero column. Therefore, $\mathbf{Q}(i)$ on the right side of the inequality also contains a non-zero column.

Observe that $\tau(n-\psi)$ is finite, and hence, $\beta^{\tau(n-\psi)}$ is non-zero. Since the non-zero terms in $\mathbf{H}[t]$ matrices are all 1, the non-zero elements in $\Pi_{t=(i-1)\tau(n-\psi)+1}^{i\tau(n-\psi)}\mathbf{H}[t]$ must each be ≥ 1. Therefore, there exists a non-zero column in $\mathbf{Q}(i)$ with all the elements in the column being $\geq \beta^{\tau(n-\psi)}$. Therefore, by Lemma 5, $\lambda(\mathbf{Q}(i)) \leq 1-\beta^{\tau(n-\psi)}$, and $\mathbf{Q}(i)$ is a scrambling matrix. □

Theorem 2 *Suppose that $G(\mathcal{V},\mathcal{E})$ satisfies the sufficient condition stated above. Algorithm 1 satisfies both the validity and convergence conditions.*

Proof: Since $v[t] = \mathbf{M}[t]\,v[t-1]$, and $\mathbf{M}[t]$ is a row stochastic matrix, it follows that Algorithm 1 satisfies the validity condition.

Using Lemma 4 and the definition of $\mathbf{Q}(i)$, and using the inequalities $\lambda(\mathbf{M}[t]) \leq 1$ and $\lambda(\mathbf{Q}(i)) \leq (1-\beta^{\tau(n-\psi)}) < 1$, we get

$$\begin{aligned}\lim_{t\to\infty}\delta(\Pi_{i=1}^{t}\mathbf{M}[i]) \;&=\; \lim_{t\to\infty}\delta\left(\left(\Pi_{i=(\lfloor\frac{t}{\tau(n-\psi)}\rfloor)\tau(n-\psi)+1}^{t}\mathbf{M}[i]\right)\left(\Pi_{i=1}^{\lfloor\frac{t}{\tau(n-\psi)}\rfloor}\mathbf{Q}(i)\right)\right)\\ &\leq\; \lim_{t\to\infty}\Pi_{i=1}^{\lfloor\frac{t}{\tau(n-\psi)}\rfloor}\lambda(\mathbf{Q}(i)) \;=\; 0\end{aligned}$$

Thus, the rows of $\Pi_{i=1}^{t}\mathbf{M}[i]$ become identical in the limit. This observation, and the fact that $v[t] = (\Pi_{i=1}^{t}\mathbf{M}[i])v[0]$ together imply that the states of the fault-free nodes satisfy the convergence condition. □